

# TOUDA知识点梳理

基于TOUDA3.0版本讲解

V2018.12.24

邓佳欣

# 目录

- TOUDA基本概念
- TOUDA开发规范
- TOUDA-IDE

# TOUDA基本概念

# 微服务

- 定义

- 围绕业务领域组件来创建应用，让应用可以独立地开发、管理和加速

- 优点

- 每个微服务组件都是简单灵活的，能够独立部署。不再像以前一样，应用需要一个庞大的应用服务器来支撑。
  - 可以由一个小团队负责，更专注专业，相应地也就更高效可靠。
  - 微服务之间是松耦合的，微服务内部是高内聚的，每个微服务很容易按需扩展。
  - 微服务架构与语言工具无关，自由选择合适的语言和工具，高效地完成业务目标即可。

# 微服务

- 设计原则

- 无状态协议**HTTP**，具备先天优势，扩展能力很强
- **JSON** 报文序列化，轻量简单，人与机器均可读，学习成本低，搜索引擎友好
- 前后端分离（**TOUDA无前端实现**，专注于服务层）

# TOUDA平台-架构

- 技术架构-SEDA(Staged Event-Driven Architecture)
  - 多服务接入、多核心服务、多主机接出
  - 服务开发、服务发布、运行管理
- 核心思想
  - 把一个请求处理过程分成几个Stage，不同资源消耗的Stage使用不同数量的线程来处理，Stage间使用事件驱动的异步通信模式
- 特点
  - 逻辑分段
  - 段与段之间队列/消息通信
  - 资源分配
- 架构原则
  - 符合微服务原则
  - 分布式服务框架
  - 平台化解决方案
  - 架构设计、业务建模、服务编排、应用发布、运行管理一体化

# TOUDA平台-架构-三段式架构

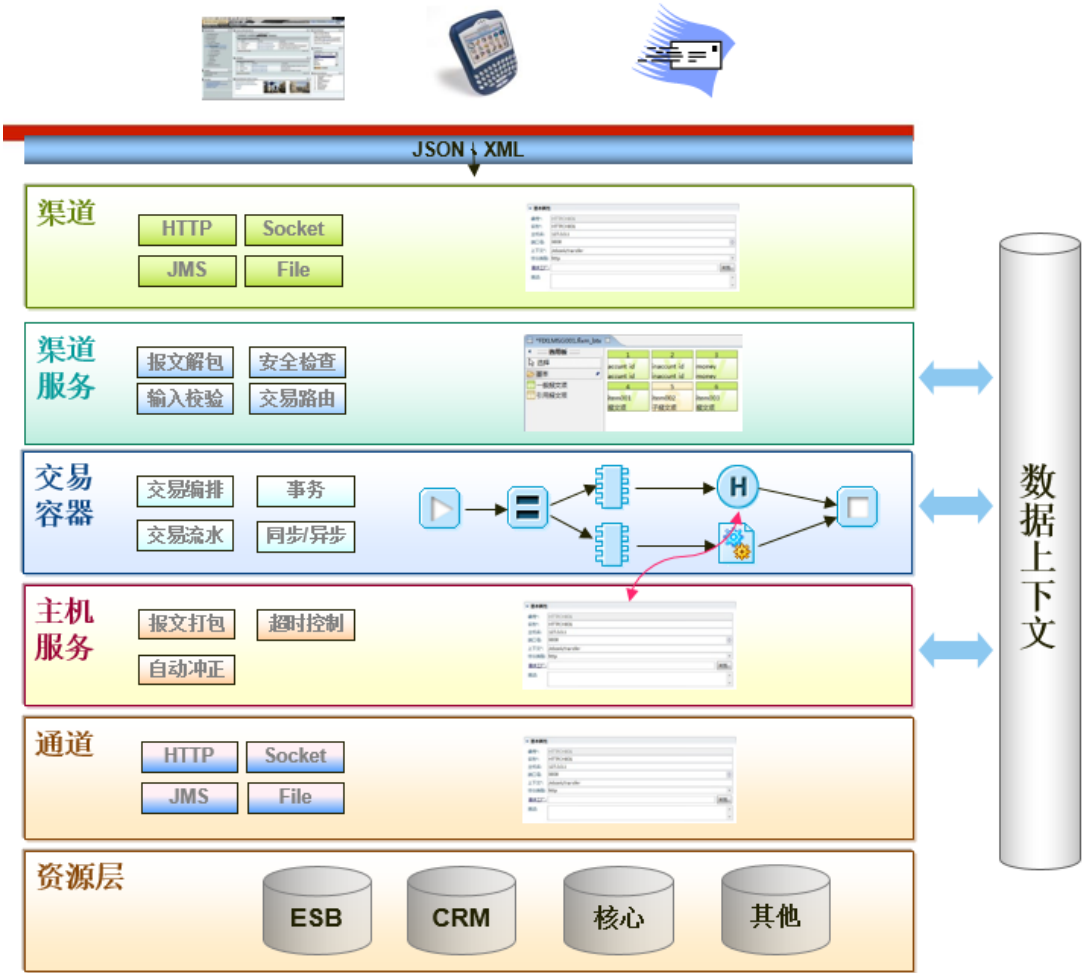
- 第一段:协议的处理、数据的处理
  - 渠道、渠道服务
  - 服务通过渠道暴露出来，所有调用touda服务的请求都必须从渠道进入。通过渠道将请求的数据传递到渠道服务（通过报文来传递），渠道服务解析请求的报文（一般支持json报文、soap报文、自定义报文），报文打/解包，将请求报文转化成一个内部识别的对象，交给第二段来处理。
- 第二段:核心业务处理
  - 交易服务、主机服务（使用较多的是交易服务）
  - 请求数据通过路由到交易服务，由交易服务处理请求，交易服务中可能涉及到调用第三方系统，此时需通过主机服务来调用（调用主机服务时同样有报文打/解包的动作），处理完后通过通道（第三段），通道处理协议层的東西。
- 第三段:服务接出
  - 通道
  - 渠道主要是接受外部调用touda接口的，即服务是通过渠道来对外暴露的，而在touda中调用第三方服务时，需通过通道来调用

# TOUDA平台-架构

- 架构目标
  - 在不确定的世界交付确定的服务
- 传统模式:一个请求开一个线程处理
  - 上下文切换
  - 资源占用
  - 锁的竞争
- 不确定的世界?
  - 外部环境因素
    - *超量请求*
    - *请求积压*
    - *并发请求*
  - 外部系统因素
    - *网络中断*
    - *服务不可用*
    - *服务超时*
- 确定的服务解决方案
  - 资源配额
  - 基于队列有策略的丢弃
  - 异步通知



# TOUDA平台-应用架构



# TOUDA平台-部署架构

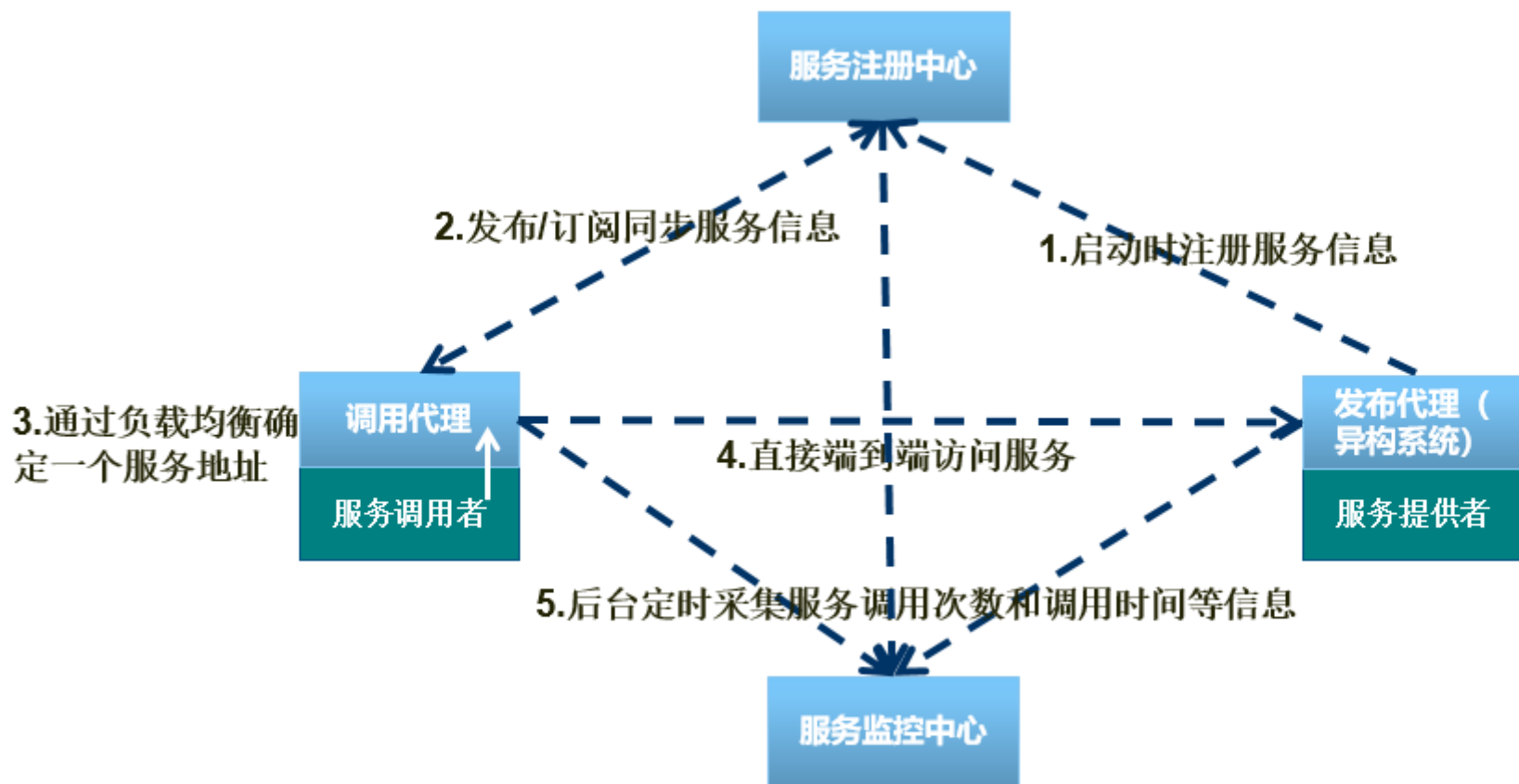
- 无容器,StandAlone模式,不依赖于tomcat等其他容器
- 只是1个JVM进程,同一台虚拟机上可以部署多个ToudaServer
- 部署/启动/停止非常简单(start.sh/stop.sh)

# TOUDA平台-功能架构-已补充



# TOUDA平台-注册中心

- 通过以下步骤达到目标:去中心化



# TOUDA平台-注册中心

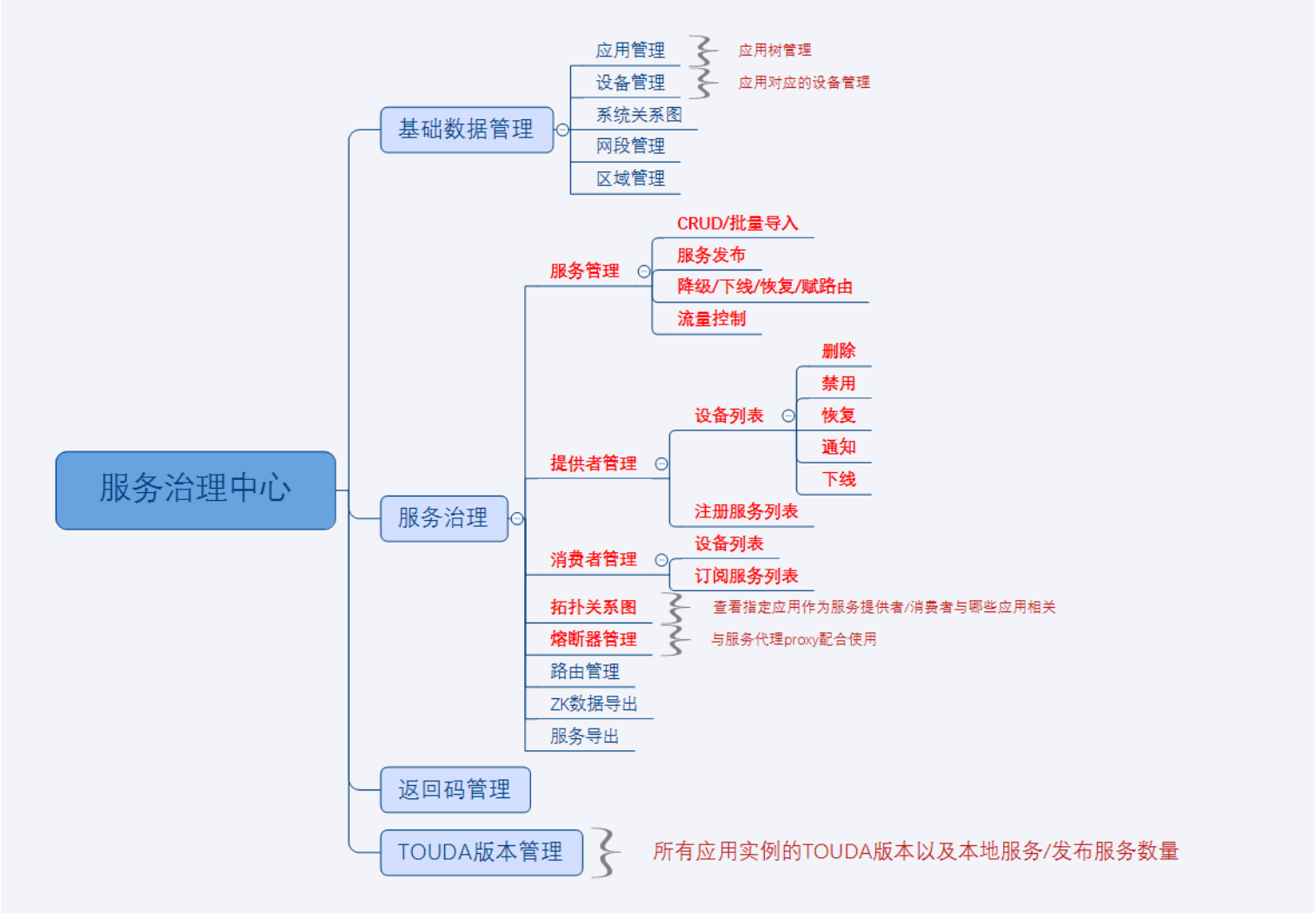
- 参与者

- **服务提供者**:应用启动时向注册中心注册服务信息
- **注册中心**:服务管理,采用zookeeper实现
- **服务调用者**:使用**调用代理**从注册中心订阅服务并缓存.并监听服务提供者信息是否发生变更,动态变更调用地址
- **服务监控中心**:接收以上三者定时发送过来的监控信息(如调用次数/调用时间),并展现

- 依赖

- 服务提供者:function.ecd.编译时会强制打入部署包
- 调用代理:com.pccc.touda.proxy/common/zookeeper.jar
- touda.properties(配置中心)
- spring-proxy.xml

# TOUDA平台-服务治理中心WEB端功能图谱



# TOUDA平台-服务治理中心&服务注册中心

- 不同之处

1. 功能不同

- a. 注册中心目前仅有服务注册以及订阅功能,主要角色为服务之间协调者.
- b. 治理中心功能请参考上文功能架构以及WEB端功能图谱,主要角色为服务之间的管理者.所有的服务发布/服务提供者消费者变更都在治理中心发生.

2. 使用者不同

- a. 治理中心主要面向运维人员
- b. 注册中心主要面向各个系统.供各个系统调用

配置中心>=测试环境的  
touda.properties



# 配置中心-设计目标

- 需要解决的问题
  - 集中管理配置项
  - 支持和多个应用共享公共配置
  - 支持发布订阅
  - 配置发布带有版本,支持回滚
  - 配置项修改实时生效
- 配置方法
  - spring-proxy.xml中增加配置项
  - touda.properties中配置项
  - 项目需依赖com.pccc.touda.zookeeper.jar **version 3.0.0+**

连接配置中心获取配置项必须指定 zk 地址、appId 和 appEnv 三个配置项,

touda.zk.address=182.180.80.102:2181↵

touda.app.id=touda\_config\_service↵

touda.app.env=dev (没有默认读取本地配置文件) ↵

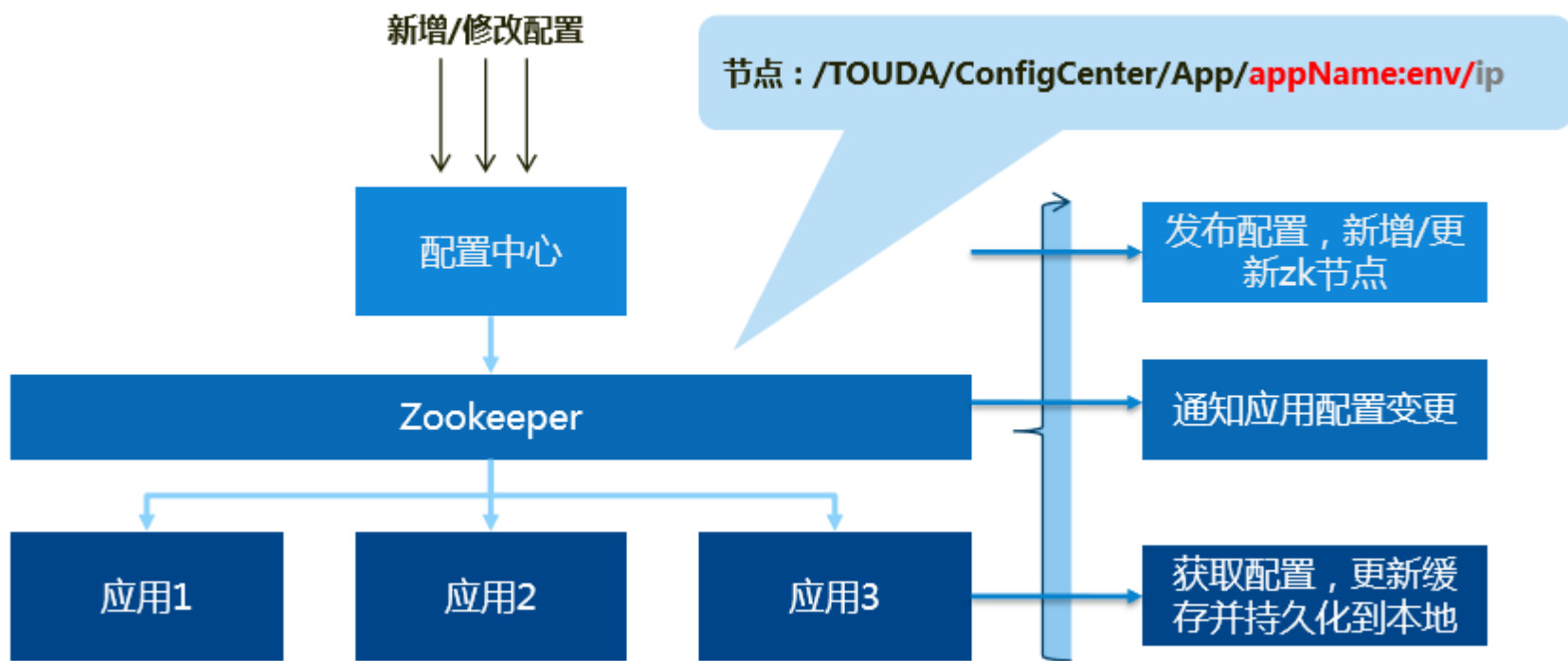
```
<!-- 若连接配置中心, 则必填; 初始化时会从配置中心, 获取配置项 -->
```

```
<!-- 不连接配置中心, 可以删除 -->
```

```
<bean id="configPreProcess" class="com.pccc.touda.zookeeper.processor.ConfigProcessor"></bean>
```

# 配置中心-技术架构

- 采用zookeeper实现
- 实时更新
- 提供web管理页面操作



# 配置中心-web管理界面-1

过滤域...  
折叠全部

展开全部

配置管理

移动互联

新一代-用户中心

新一代-活动中心

新一代-权益中心

新一代-订单中心

新一代-交互网关

新一代-基础数据服务平台

新一代-风控平台

新一代-运营运维平台

新一代-新商城

新一代-客服平台

微信小程序

智能营销系统

销售管理系统PDA

TOUDA

dfs

测试

管理

类型: 全部 名称: 请输入名称... 环境: uat 查询 重置

新增 修改 删除 清空选择 配置项管理 提示: 双击单条记录可以快速查看明细!

<input type="checkbox"/>	名称	类型	环境	关联项目	发布版本	负责人	描述
<input type="checkbox"/>	piap_srh_mgmt_web	应用	uat	1454	release-201806...	hehaoxiang	
<input type="checkbox"/>	piap_uinfo_service	应用	uat	1454	release-201806...	xz_hanguoq...	用户中心, 用户模块 ...
<input type="checkbox"/>	piap_uacc_service	应用	uat	1454	release-201806...	xz_hanguoq...	用户中心-用户模块账...
<input type="checkbox"/>	piap_uenc_service	应用	uat	1454	release-201806...	xz_hanguoq...	用户中心-用户模块密...
<input type="checkbox"/>	piap_bauth_service	应用	uat	1454	release-201805...	xz_wangnan	用户中心-鉴权模块服...
<input type="checkbox"/>	piap_oauth_service	应用	uat	1454	release-201805...	xz_yanglong...	用户中心-单点登录模...
<input type="checkbox"/>	piap_sso_hweb	应用	uat	1454	release-201806...	xz_nishengl...	用户中心-用户模块单...
<input type="checkbox"/>	piap_sso_mweb	应用	uat	1454	release-201805...	xz_nishengl...	用户中心-用户模块单...
<input type="checkbox"/>	piap_oauth_mana...	应用	uat	1454	release-201806...	xz_yanglong...	用户中心-用户模块单...
<input type="checkbox"/>	piap_bauth_mana...	应用	uat	1454	release-201805...	xz_wangnan	

# 配置中心-web管理界面-2

piap\_oauth\_service(uat)配置项管理

发布

回滚

发布历史

返回项目管理

表格

文本

更改历史

实例列表

公共配置项

绑定服务

新增

修改

删除

每页 20 条记录

<input type="checkbox"/>	发布状态	类型	Key	Value
<input type="checkbox"/>	已发布	环境变量	touda_app_id	piap_oauth_service
<input type="checkbox"/>	已发布	环境变量	DATABASE_DRIVER_CLASS	com.ibm.db2.jcc.DB2Dri
<input type="checkbox"/>	已发布	环境变量	DATABASE_GROUP_NAME	default
<input type="checkbox"/>	已发布	环境变量	DATABASE_JDBC_TYPE	IBMDB2Driver(Type4)
<input type="checkbox"/>	已发布	环境变量	DATABASE_MAX_POOL_SIZE	50
<input type="checkbox"/>	已发布	环境变量	DATABASE_MIN_POOL_SIZE	10
<input type="checkbox"/>	已发布	环境变量	DATABASE_PASSWORD	U2FsdGVkX1/X6M3uLy
<input type="checkbox"/>	未发布	环境变量	DATABASE_POOL_SIZE <div>改</div>	20
<input type="checkbox"/>	已发布	环境变量	DATABASE_URL	jdbc:db2://182.180.198.
<input type="checkbox"/>	已发布	环境变量	DATABASE_USERNAME	usr_user
<input type="checkbox"/>	未发布	一般变量	force.request.with.env <div>新</div>	false

修改

关联项目

piap\_mall\_common\_config(uat),...

关联服务

redis-mall-1 ▾

负责人\*

cuiyongzhi

描述

请输入描述...

取消

保存

# 配置中心-功能

- 管理配置项(应用配置/公共项配置)
  - 一般变量(##0):将会写在touda.properties中
    - pay.merchant.no=abc1234##0
  - 环境变量(##1):将会初始化在云上容器内的环境变量(如下图)
    - touda\_app\_env=uat##1

```
vcap@87f01508-0220-4a0d-7300-f212:~$ env
CF_INSTANCE_ADDR=182.180.118.108:61012
TERM=vt100
force.subscribe.service=true
touda_app_env=uat
DATABASE_PASSWORD_1=U2FsdGvkX19NAmtpbpt7VHw9fj11wenohe/Ad6U=
DATABASE_PASSWORD_2=U2FsdGvkX1/U+qTUya/xRDwdDG2PoBNRjkSTQks=
DATABASE_PASSWORD_3=U2FsdGvkX19Nj115APU6zBeUX5RIN6EzseKdpDo=
VCAP_APP_PORT=8080
USER=vcap
```

- 可在本页面绑定云服务
- 配置项在发布后生效
- 查看已读取该配置的应用列表
- 回滚参数
- 代码使用API: *ConfigUtils.getProperty (key) /ConfigUtils.getVCAPService(key)*

# 熔断器

- 概念
  - 在特定的场景下关掉当前的通路,从而起到保护整个系统的效果
  - 参考@生活中的保险丝
- 熔断条件
  - 断路器打开
  - 线程池满
- 熔断效果:执行服务降级,不再调用远程服务,返回快速失败报文
- 集成在com.pccc.touda.proxy.jar中
- 待跟进:服务治理中心的web端熔断器管理配置页面与此处的关系

# 熔断器-web端管理页面

熔断器修改

熔断器\*

update

熔断组\*

update

熔断返回

是

开启超时

是

断路请求数量/秒

线程池核心数量

队列等待长度

熔断请求并发量/秒

共享

是

熔断器状态

开启

线程池名称\*

update

开启断路

是

超时时间

以ms为单位

断路错误阈值/秒

百分比0到100的数字

线程池最大数量

请求日志摘要开关

是

熔断时长

以ms为单位

说明

熔断返回值为服务管理中服务的降级取值

取消

保存

# ToudaDFS-分布式文件系统(即图片服务器)

- 设计背景

- 上PaaS云硬性要求:云上容器不能保留任何文件
- 替换传统的挂载NAS盘方案的性能低及管理问题

- 设计目标

- 海量小文件存储
- 高并发访问
- 实现了软件方式的RAID,提高磁盘的利用率
- 图片服务器
- 线性扩容,在线扩容

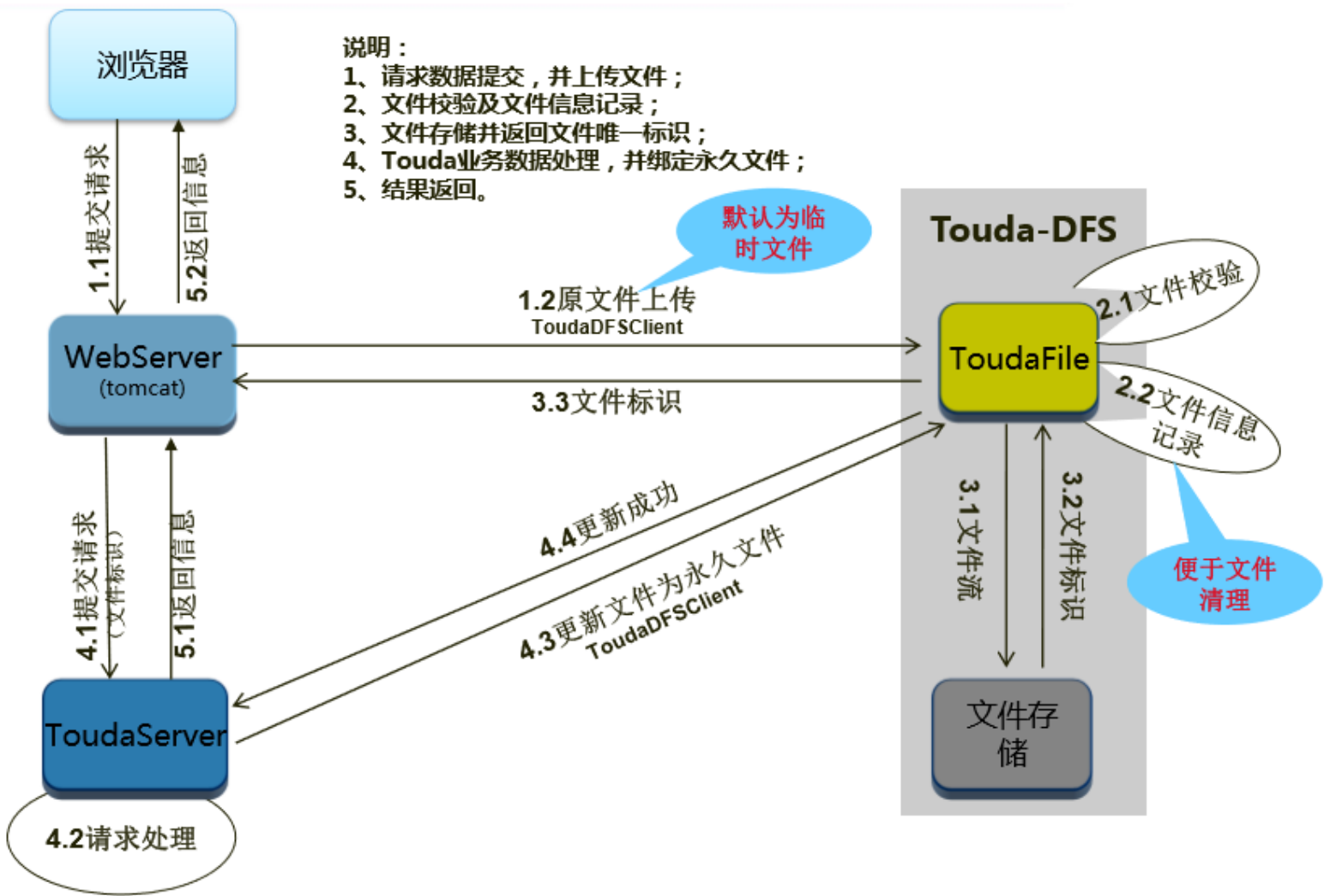


# ToudaDFS-分布式文件系统(即图片服务器)

- 文件上传流程

- 1、请求数据提交，并上传文件；
- 2、文件校验及文件信息记录；
- 3、文件存储并返回文件唯一标识；
- 4、Touda业务数据处理，并绑定文件；
- 5、结果返回

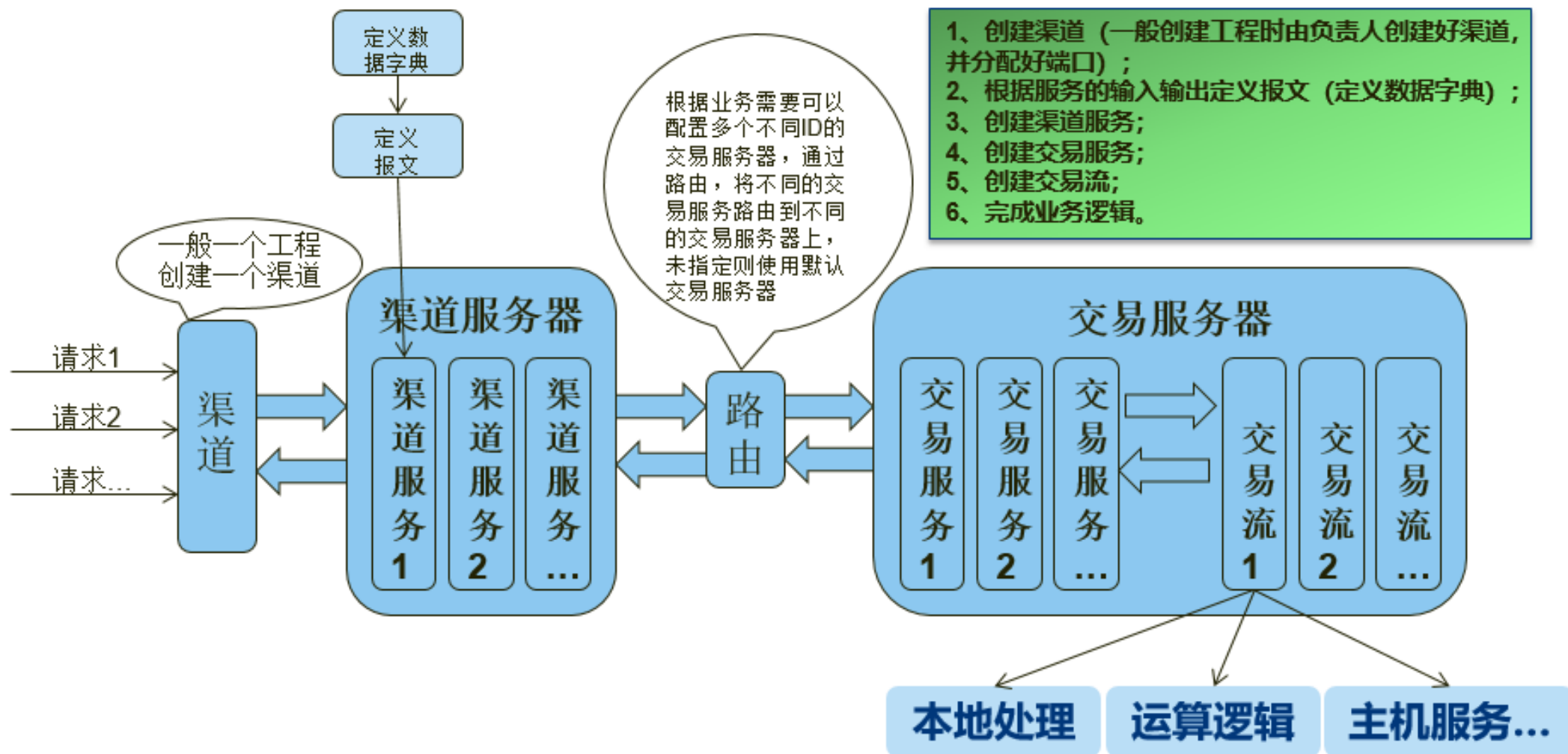
# ToudaDFS-分布式文件系统(即图片服务器)



# ToudaDFS-注意点

- 文件默认临时保留30天,调用绑定文件API可以更新为永久文件.
- 根据业务的需求, 进行清理, 支持按组、月、按年清理
- API在com.pccc.touda.dfs.client.ToudaDFSClient.java
  - 单文件上传(upload)
  - 批量文件上传(uploadMutliFiles)
  - 上传本地文件(uploadLocalFile)
  - 下载文件(download)
  - 下载文件并写到本地(downloadAndWriteLocal)
  - 绑定文件(bindingFile)
  - 批量绑定文件(bindingFiles)
  - 删除文件(deleteFile)
  - 批量删除文件(deleteFiles)
  - 获取展示信息(getShowUrl)
- dfsFileid/fileid是以上API的核心,各个系统需要自行存储
  - dfsFileid示例:office01/a/b/c.jpg
  - fileid示例:uuid
- 常用的单文件上传的API中需设置文件是否可外网访问(选填),是否为永久文件(选填),以及渠道服务号(必填)
- 需配置touda.file.direct/ touda.file.server.ip配置项
- 需要填写05表申请以下渠道服务以及文件存储策略(策略目前需线下邮件申请)
  - com.pccc.touda.toudafile.servicecustom.J\_UPLOAD\_001\_0001
  - com.pccc.touda.toudafile.servicecustom.J\_FILE\_003\_0001
  - com.pccc.touda.toudafile.servicecustom.J\_FILE\_003\_0002
  - com.pccc.touda.toudafile.servicecustom.J\_FILE\_003\_0003

# TOUDA-profile-数据流示意



# TOUDA-渠道

- 概念

- 处理协议层的东西，接受外部调用，支持HTTP、socket、JMS、File
- 已有HTTP渠道/KAFKA渠道/MQ渠道
- 渠道提供一种方式将Touda提供的核心服务通过HTTP/TCP协议暴露出去，从而允许第三方应用对Touda进行访问

- 注意点

- 每个应用只能有1个HTTP渠道,必须为8080(上PaaS云要求),避开9090(被function占用)
- 每个应用可以有其他渠道
- 命名规范:通讯协议(大写)+\_+业务域（三位字符，不得超过5位）+\_+序号（二位数字），全部使用大写字母，序号为递增，字符长度不能超过20个字符，如：  
HTTP\_PUB\_01
- 扩展实现类以Transport结尾
- 注册拦截器类名必须以Launch结束

# TOUDA-渠道-HTTP渠道示意

HTTP\_MALL\_01.chh\_btx

HTTP渠道

基本属性

编号\*:

HTTP\_MALL\_01

名称\*:

HTTP\_MALL\_01

主机名:

0.0.0.0

端口号:

8080

上下文\*:

/piap\_mall\_markert\_service

协议类型:

http

请求工厂:

浏览...

描述:

渠道处理器

最小实例数:

50

最大实例数:

100

实例递增数:

1

实例空闲时间(s):

60

处理器获取超时(s):

5

处理器工厂:

浏览...

高级属性

数据转换:

浏览...

扩展实现:

浏览...

扩展属性

属性名

属性值

描述

ENCODING

UTF-8

# TOUDA-渠道-KAFKA渠道示意

## 自定义渠道

### 基本属性

编号\*: KAFKA\_MALL\_01

名称\*: KAFKA\_MALL\_01

描述: 消费活动中心通过KAFKA发出的活动终止通知

### 渠道处理器

最小实例数: 50

最大实例数: 50

实例递增数: 1

实例空闲时间(s): 60

处理器获取超时(s): 5

处理器工厂:  浏览...

### 高级属性

扩展实现\*: com.pccc.touda.kafka.transport.cust.KafkaTransport

浏览...

### 扩展属性

属性名	属性值	描述
TransCode	J_MALL_002_0054	渠道号
BOOTSTRAP_SERVERS_CON...	\${kafka1.bootstrap_servers...	kafka地址
Topic	\${kafka1.topic}	kafka的topic
ENABLE_AUTO_COMMIT_C...	\${kafka1.enable_auto_com...	是否自动提交
FETCH_MAX_BYTES_CONFIG	\${kafka1.fetch_max_bytes_c...	每次最多获得数据为多少kb

# TOUDA-渠道-KAFKA渠道注意项

KAFKA渠道扩展属性			
key	Value(示例)	必填	description
BOOTSTRAP_SERVERS_CONFIG	\${kafka.servers}	是	kafka brokers地址
GROUP_ID_CONFIG	\${groupName}	是	kafka consumer group
Topic	\${topicName}	是	kafka topic
ENABLE_AUTO_COMMIT_CONFIG	FALSE	否	是否自动提交，不填则默认为false
FETCH_MAX_BYTES_CONFIG	1024	否	每次最多获得数据为多少kb，不填则默认为1M
waitSec	10	否	无数据默认等待时间，单位（S），不填则默认为5S
maxCurrentFlowCount	50	否	控制并发量，不填默认为100
TransCode	J_TES_001_0001	否	处理kafka消息的渠道服务，不填写默认取请求报文Head头中的d_transcode



# TOUDA-渠道-MQ渠道示意

MQ\_PCS\_01.chctm\_btx

自定义渠道

基本属性

编号\*: MQ\_PCS\_01

名称\*: MQ\_PCS\_01

描述:

渠道处理器

最小实例数: 50

最大实例数: 50

实例递增数: 1

实例空闲时间(s): 60

处理器获取超时(s): 5

处理器工厂:

高级属性

扩展实现\*: com.primeton.btp.spi.transport.cust.mq.WebsphereMqCus

扩展属性

属性名	属性值	描述
host	182.180. [REDACTED]	
port	10100	
channelName	PMS.SVRCONN	
queueManagerNa...	QM_PMS	
ccsid	1208	

# TOUDA-渠道-MQ渠道注意项

- REQUEST\_CREATOR设为： com.primeton.btp.spi.transport.cust.mq.WebsphereMqCustRequestCreator

WebSphereMQ渠道扩展属性		
配置项	描述	默认值
host	MQ服务器地址	
port	MQ监听端口	1414
channelName	MQ通道名	
queueManagerName	MQ队列管理器名	
ccsid	MQ消息体字符编码	1208
batchCount	单次从MQ中取消息的条数	20
requestQName	MQ队列名	
maxCurrentFlowCount	最大并发线程数	
REQUEST_CREATOR	MQ渠道请求创建类	
IS_RECORD_MESSAGE	是否记录系统日志	TRUE
TransCode	处理MO消息的渠道服务	

# TOUDA-通道

- 概念

- 与渠道对应,处理协议层的东西，调用外部服务
- 通道提供一种方式允许Touda通过HTTP/TCP协议去访问第三方提供的服务

- 注意点

- 命名规则:通讯协议(大写)+\_GW\_+目标主机简称，全部使用大写字母，字符长度不能超过20个字符，如：HTTP\_GW\_GTP
- 扩展实现类必须以Endpoint结束
- 注册拦截器类名必须以Launch结束

# TOUDA-通道示意

HTTP\_GW\_GTP.gwh\_btx

## HTTP通道

### 基本属性

编号\*: HTTP\_GW\_GTP

名称\*: HTTP\_GW\_GTP

请求URL\*: http://\${host}

描述: GTP通道

### 线程池

最小线程数: 50

最大线程数: 50

存活时间(s): 60

队列大小: 0

拒绝策略: 浏览...

### 通道访问器

最小实例数: 50

最大实例数: 50

实例递增数: 1

实例空闲时间(s): 60

访问器获取超时(s): 5

访问器工厂: 浏览...

### 高级属性

数据转换: 浏览...

扩展实现: 浏览...

### 扩展属性

属性名	属性值	描述

# TOUDA-渠道服务

- 渠道服务

- 将交易服务、主机服务挂在指定的渠道上，将渠道接收的数据转换后转发到对应的服务上
- 报文打/解包（请求进来时报文解包，响应时报文打包）、安全检查、输入校验、交易路由

- 注意点

- 命名规范:
  - 渠道分类：渠道简称(J/X)+\_+业务域简称(三位, 不得超过5位), 全部使用大写字母, 字符长度不能超过20个字符, 如: J\_PUB
  - 渠道服务名称：渠道(J/X)+\_+交易码, 字符长度不能超过20个字符, 如: J\_PUB\_001\_0001
- 渠道服务的描述必须填写, 描述信息-查询/修改/新增/删除, 如: 用户信息查询-查询
- 一个渠道服务就是一个对外接口.
- 渠道服务:交易服务:交易流:请求报文:响应报文=1:1:1:1:1
- 渠道服务在服务定制目录下面
- 扩展实现类必须以TransportService结束

# TOUDA-渠道服务示意

J\_MALL\_002\_0001.chs\_btx

MALL\_002\_0001.process\_btx

### 渠道服务

基本属性

服务码\*: J\_MALL\_002\_0001

服务名\*: J\_MALL\_002\_0001

目标渠道\*: HTTP\_MALL\_01

服务类型: 交易服务

服务代码\*: TS\_MALL\_002\_0001 浏览...

服务功能码: MALL\_002\_0001

单向: ☐

处理类型: 请求

请求报文: J\_MALL\_REQ\_002\_0001 浏览...

响应报文: J\_MALL\_RES\_002\_0001 浏览...

描述: 查询活动列表

高级属性

超时(s): 0

扩展实现: 浏览...

扩展属性

属性名	属性值	描述

# TOUDA-交易路由

- 概念

- 渠道服务器到交易服务器，对核心服务可以规划出独立的线程池来处理，也可以将核心的服务指向到单独的交易服务器来处理
- 根据业务需要可以配置多个不同ID的交易服务器，通过路由，将不同的交易服务路由到不同的交易服务器上，未指定则使用默认交易服务器
- 参考[TOUDA-profile-数据流示意](#)中交易路由的功能示意

# TOUDA-交易服务

- 概念

- 交易服务提供了处理用户业务能力的逻辑。支持事务、冲正、流水、超时管理能力。交易服务挂在渠道服务上，可以调用本地处理、主机服务、冲正服务能力
- 交易服务器为交易服务提供了队列缓冲能力
- 交易服务通过匹配的交易流进行**业务编排**

- 注意点

- 命名规范

- 交易服务分类名称：业务域简称，全部使用大写字母，字符长度不能超过20个字符，如：PUB。
    - 交易服务名称：TS+\_+交易码，字符长度不能超过20个字符，如：**TS\_PUB\_001\_0001**
    - 扩展实现类必须以TransService结束



# TOUDA-交易服务示意

● TS MALL 002 0001.txs btx ✕

## 交易服务

### ▼ 基本属性

服务码\*: TS MALL 002 0001

服务名*:	TS MALL 002 0001
-------	------------------

交易类型:

事务类型: 无事务

### 交易流程\*:

[浏览...](#)

功能码	交易流程
MALL_002_0001	com.pccc.mall.mall_markert_service.servicecustom._process.ACTIVITY.M...

描述:

**▼ 高级属性**

超时(s): 0

记录业务流水: ☐记录系统日志: ☐

### 扩展实现:

浏览...

### ▼ 扩展属性

[illegible]

# TOUDA-事务

- 事务中分为两类：全局事务、本地事务和无事务。

- 全局事务：系统由业务活动完成，在业务活动中可以加入原子工作项，每一原子工作项（在TOUDA中原子工作项就是流程步：例如本地处理、主机服务图元）都会包含一事务相关的上下文以及当前事务所处的状态，在原子工作项加入到业务活动时，原子工作项相关数据都存入数据库中以便业务活动中原子工作项的回滚。在业务活动中的每一原子工作项都成功时，完成业务活动的提交，如果其中一原子工作项发生异常，则要把已经完成的原子工作项执行回滚操作。
- 本地事务：由数据库事务进行控制，通过数据库连接的进行统一提交或者回滚。
- 无事务：在无事务模式下，TOUDA平台不提供任何的事务控制，完全有业务开发人员来保证业务的一致性。

- 注意点:

- 目前开发规范限制,支持但是禁止使用全局事务
- 查询类及无事务要求的交易,必须选择无事务

# TOUDA-主机服务

- 概念

- 主机服务是外系统提供服务的本地代理，例如，银行代缴水电的业务中，需要调用电力公司的欠费查询服务，这样代缴水电的业务系统中，需要定义一个相应的“欠费查询”的主机服务，以供相应的交易进行集成

- 主要工作

- 1、系统间的数据（格式）转换，如需要将水电代缴系统内部的数据（格式），转换为电力公司业务系统所需要的数据（格式）；
  - 2、通过指定的协议通道发送到外部系统：如需要通过TCP的协议通道发送到电力公司；
  - 3、异常处理，请求外系统服务发生异常时，进行处理

- 注意点

- 命名规范:
    - 主机服务分类目录：报文类型简称(J/X)+目标系统的简称，全部使用大写字母，字符长度不能超过20个字符，如： X\_GTP。
    - 主机服务名称：由公共组统一生成
  - 扩展实现类必须以HostTran结束

# TOUDA-主机服务示意

KAFKA\_MALL\_01.chctm\_btx

HTTP\_MALL\_01.chh\_btx

HTTP\_GW\_GTP.gwh\_btx

GTPQueryInfor.hts\_btx

主机服务

基本属性

服务码\*:GTPQueryInfor

服务名\*:GTPQueryInfor

冲正服务码:

目标通道\*:HTTP\_GW\_GTP

单向:☐

请求报文:J\_MALL\_REQ\_002\_0001

响应报文:J\_MALL\_RES\_002\_0001

描述:这是一个demo的主机服务,实际上服务名称应该由公共组统一生成

高级属性

超时(s):0

超时动作:忽略

扩展实现:

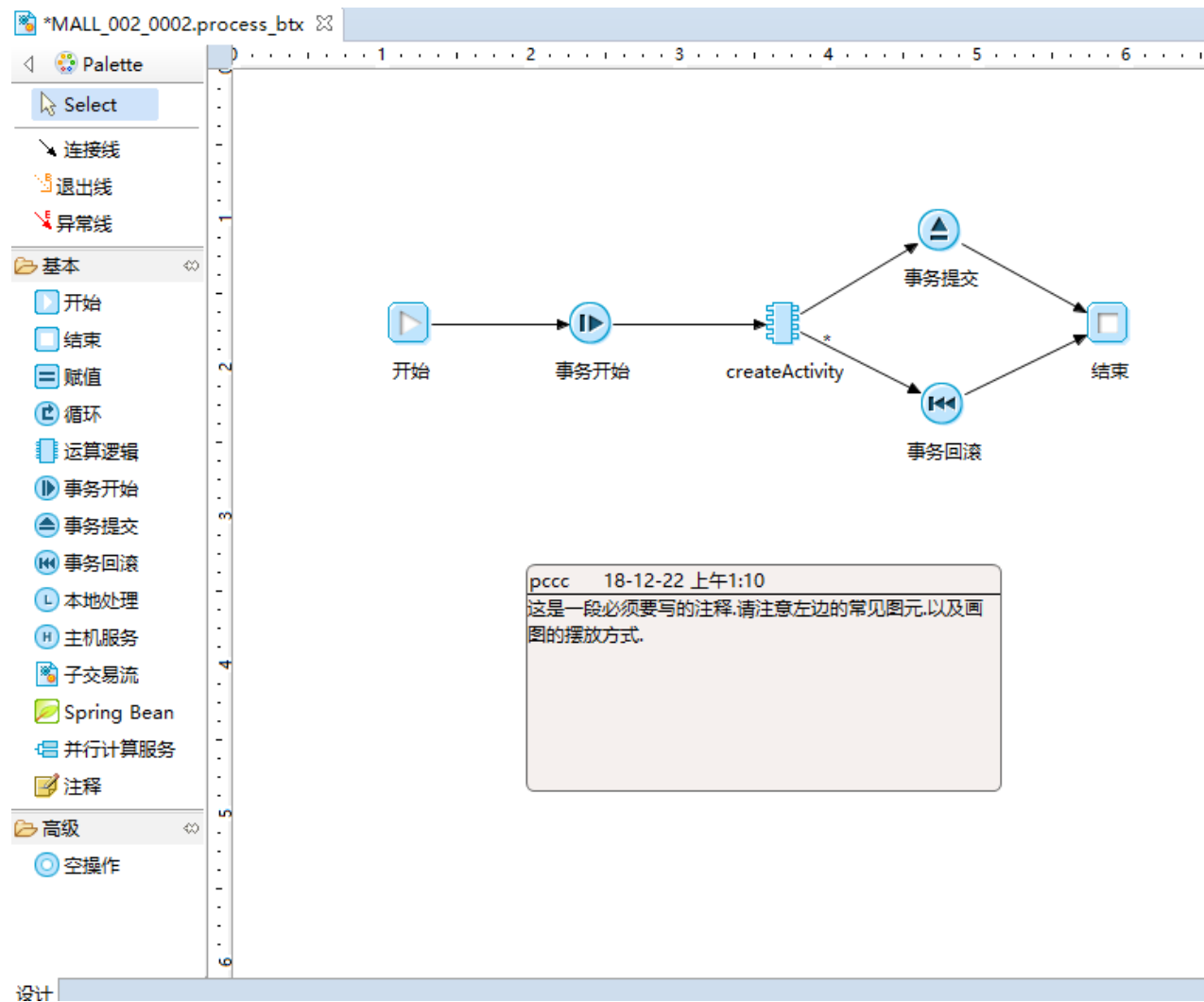
扩展属性

属性名	属性值	描述
CUSTOM_TIMEOUT_ACTION_...		超时处理类, 必须实现接口com.primeton....

# TOUDA-交易流

- 概念
  - Touda产品的一个核心组成部分，用图形化的方式来描述处理交易流程，即用“画图”的方式来“写代码”。
  - 每一步可独立执行的操作叫做图元
- 好处
  - 交易流程直观清晰；
  - 可以拖拽得方式来使用其他构件（包括基础构件库）提供的服务，面向构件不仅落到实处，而且便于理解和掌握
- 注意点
  - 命名规范
    - 交易流分类：功能模块简称，使用有意义的英语单词简写，全部使用大写字母，字符长度不能超过20个字符，如：PUB。
    - 交易流名称：交易码，如：PUB\_001\_0001
  - 变量定义
    - 交易流中的变量可以是java基础类型、List、Map，自定义类型等。
    - 变量的命名规则：使用有意义的英文单词简写，首个英语单词首字母用小写,其他单词首字母用大写字母开头，其余用小写字母。如：  
custName
  - 交易流程设计
    - 交易流程组装的构件统一从左至右按流程进行摆放，摆放需要整齐，容易看懂。
    - 交易流程必须采用注释构件进行注释，交易流程中的循环、并发操作也必须需要进行注释。
    - 交易流程的并发图元内的构件必须从上到下进行摆放。
    - 对结点进行注释时，选择注释图元按“C”用虚线连接。

# TOUDA-交易流示意



# TOUDA-数据-命名SQL

- 概念

- 主要指【服务定制】目录下的【数据】目录下的命名SQL
- 本质上就是iBatis（MyBatis的前身）的SQL mapper文件
- 命名SQL集是一系列命名sql的集合，用来统一管理命名sql。它是给一个sql起一个唯一标识，并指定如何处理sql所需要的参数和执行完后的结果。它可以在参数和返回值中使用实体，用户可以用命名sql执行对应sql的全部操作

- 注意点

- 除【架构设计】目录下没有【数据】目录之外，其他3个构建包都含有
- 命名规范
  - 命名Sql文件名：功能模块简称+SqlMap，如：CustInfoSqlMap
  - Sql Id：操作(add、modify、remove、query)+自定义如：queryCustAccoutInfoSqlName
  - 必须使用TOUDA向导创建命名Sql文件

# TOUDA-数据-命名SQL示意

```
ActivitySqlMap.namingsqlx
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- author:pccc -->
3 <sqlMap>
4   <parameterMap class="commonj.sdo.DataObject" id="parameterMap">
5     <parameter javaType="date" jdbcType="DATE" property="dateType"/>
6   </parameterMap>
7   <resultMap class="commonj.sdo.DataObject" id="resultMap">
8     <result column="TYPEID" javaType="string" property="typeId"/>
9   </resultMap>
10  <select id="addActivity" parameterMap="parameterMap" resultMap="resultMap">
11    INSERT INTO TABLE VALUES({id},{value})
12  </select>
13 </sqlMap>
```



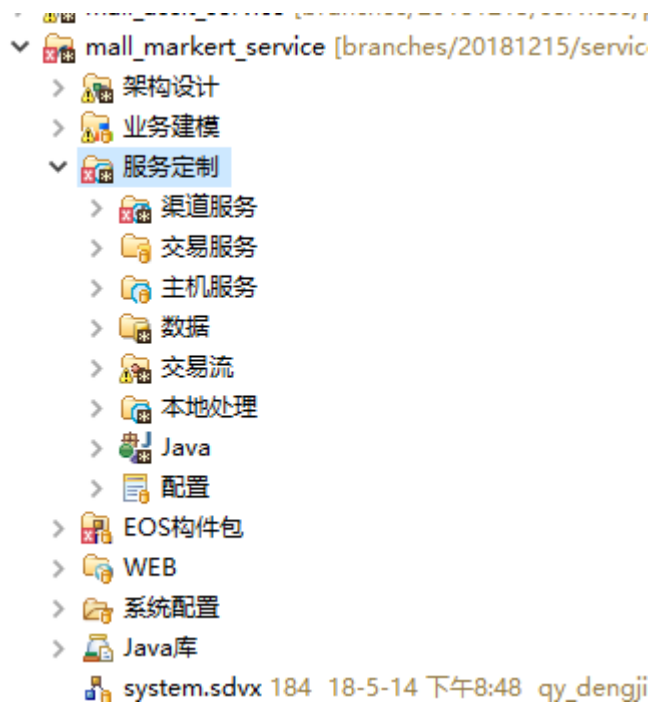
# TOUDA-本地处理

- 概念:

- 实现特定接口的Java类.为用户提供了本地业务处理能力的Java代码类

- 注意点

- 本地处理创建的位置必须在服务定制目录下的本地处理下
  - 可以由交易流直接调用
  - 本地处理名称: 英语单词首字母大写,其他字母为小写+Local, 如: ExcuteSqlLocal



# TOUDA-业务建模-数据类型

- 概念

- 业务建模主要有数据类型、数据字典、报文定义等七部分组成
- 主要描述业务模型

- 数据类型

- Touda内置提供了9种数据格式，包括
  - 布尔型:基于java.lang.Boolean
  - 整型:基于java.lang.Integer
  - 浮点型:基于java.lang.Float
  - 双精度型:基于java.lang.Double
  - 字符串型:基于java.lang.String
  - 日期型:基于java.util.Date
  - 时间型:基于java.util.Date
  - 大数字型: 基于java.math.BigDecimal
  - 金额型:基于java.lang.Float
  - 自定义数据类型

# TOUDA-业务建模-数据字典

- 数据字典

- 元数据是描述数据的数据，数据字典是数据描述的基本元素，也是元数据的基本内容
- 命名规范
  - 数据字典分类目录：设定为项目名称。
  - 文件名：META\_DICT
  - 元数据：使用有意义的英文单词简写，名字必须以一个字母开头，后面的单词用大写字母开头，字符长度不能超过20个字符，如：custName

# TOUDA-业务建模-报文定义

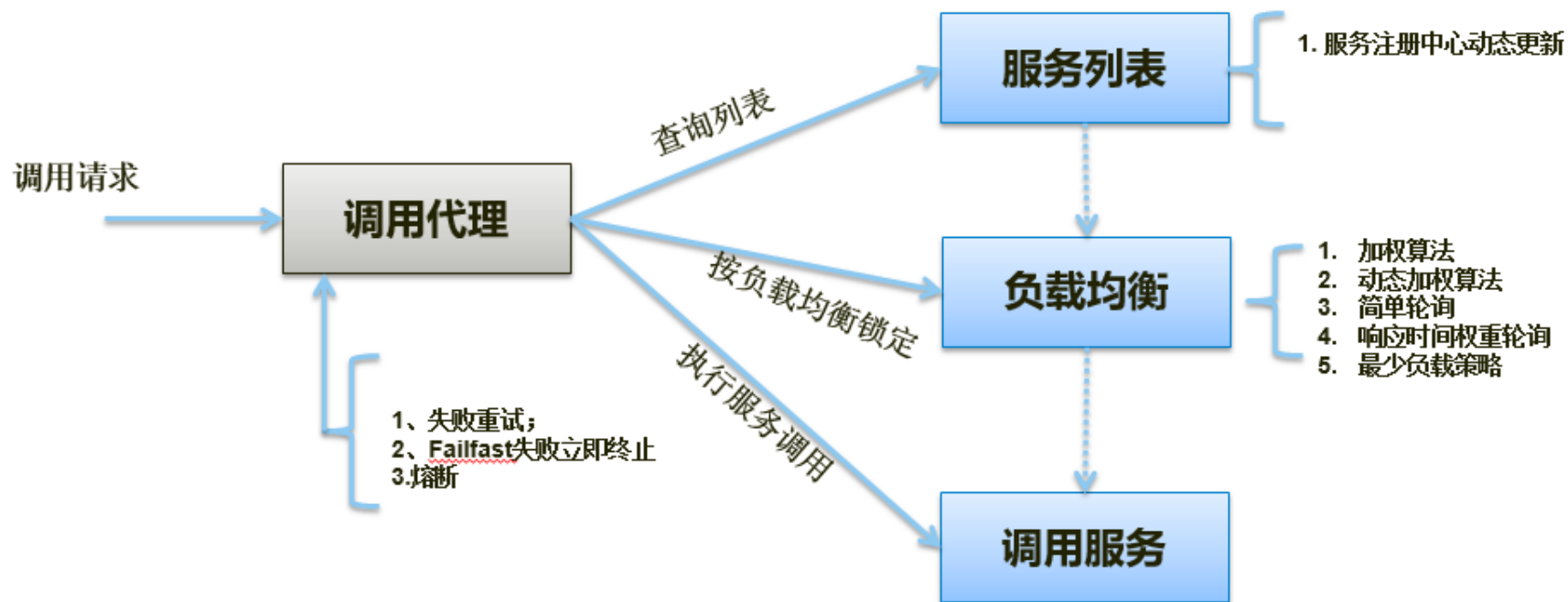
- 报文定义

- 报文是数据传播的载体，外部系统将报文封装在请求数据中，传输到Touda中，Touda将完成业务操作的内部数据转换成报文，封装到响应数据中，传输给外部系统，完成系统的交互

- 命名规范

- 渠道报文：报文类型简称(J/X)+\_+REQ/RES(请求/响应)+\_+ [交易码](#)，字符长度不能超过20个字符，如：J\_REQ\_PUB\_001\_0001
- 通道报文：统一由**公共组**生成
- 渠道报文分类目录：报文类型简称(J/X)+\_+业务域，全部使用大写字母，字符长度不能超过20个字符，如：J\_PUB,X\_PUB
- 通道报文分类目录：报文类型简称(J/X)+目标系统的简称，如：X\_GTP
- 渠道响应报文中必须包含**returnCode**和**returnMsg**, returnCode返回000000代表成功,其他情况returnCode定义参见返回码定义规范

# TOUDA-服务代理示意图



- 1、Touda代理相关jar包;
- 2、touda.properties;
- 3、spring-proxy.xml.

# TOUDA-服务代理

- 概念
  - 调用Touda服务的代理实现
  - 任何需要调用Touda服务的应用都必须集成,主要为com.pccc.touda.proxy.jar包
- 注意点
  - Touda3.0版本新增了熔断器功能
  - **服务订阅/负载均衡/失败策略/熔断器**也在此处实现,当然,**调用其他服务**更是其本能工作必须有的2个配置文件:spring-proxy.xml/touda.properties
- 通过Invoke调用指定服务（以下方法皆有多态实现）
  - **同步调用**: `Invoker.invoke(String serviceName, Map<String,Object> headerMap, Object value)`
  - **异步调用**: `Future<Result> asynInvoke(String serviceName, Map<String,Object> headerMap, Object value)`

# TOUDA-服务代理-spring-proxy.xml

- TOUDA底层采用了**部分spring代码**，该配置文件**约等于**applicationContext.xml
- 配置服务调用com.pccc.touda.common.ServiceConfigManager
  - 配置长这样\${connect\_flag},service\_full\_name,service\_default\_url
  - connect\_flag=2代表直连,1代表注册中心订阅。**DEV/SIT环境填2,UAT/PROD填1.**
  - service\_full\_name长得像com.pccc.touda.monitor.servicecustom.J\_MON\_001\_0001
  - service\_default\_url就是一个普通的URL，代表直连时默认调用地址。可以带参数。只有在直连时才生效
  - J\_MON\_001\_0001代表监控中心使用的服务，雷打不动的必须添加，不分环境必须为直连
- 配置使用配置中心com.pccc.touda.zookeeper.processor.ConfigProcessor
- 配置使用云上Redis服务com.pccc.touda.common.preprocess.JedisClusterBeanProcess
- 参考一个实际上使用的spring-proxy.xml



XML源文件

# TOUDA-服务代理-touda.properties-1

- 应用声明:touda\_app\_id/touda.app.id=piap\_system\_module\_service
- 环境声明:touda\_app\_env/touda.app.env=dev/sit/uat/prod/press
- 监控上报标识:monitor.report.flag/monitor\_host
- ZK地址:touda\_zk\_address/touda.zk.address=ip:port...
- 配置中心地址:cfg\_host=ip
- 日志输出到日志平台:kafkaAppender.services+kafkaAppender.topic
- 服务调用方式(直连/订阅):touda.connectmethod+force.subscribe.service



# TOUDA-服务代理-touda.properties-2

- 配置项key命名规范

key中的命名严禁使用拼音与英文混合的方式，更不允许直接使用中文的方式。key只能包含数字、字母、.和下划线

说明：正确的英文拼写和语法可以让阅读者易于理解，避免歧义。注意，即使纯拼音命名方式也要避免采用。

正例：alibaba / taobao / youku / hangzhou 等国际通用的名称，可视同英文。

反例：DaZhePromotion [打折] / getPingfenByName() [评分] / int 某变量 = 3

- 配置项key命名分割

key中的命名涉及某个功能或者某个组件的多个配置，**一般变量采用"."进行分割，环境变量采用"\_"分割**，对于无法分割的词语，使用驼峰风格，遵从驼峰形式

正例：marcoPolo /touda.proxy.server.port

- 配置项value 换行问题

如果配置项的value需要换行，但请使用\进行连接

key=value\value

# TOUDA开发规范

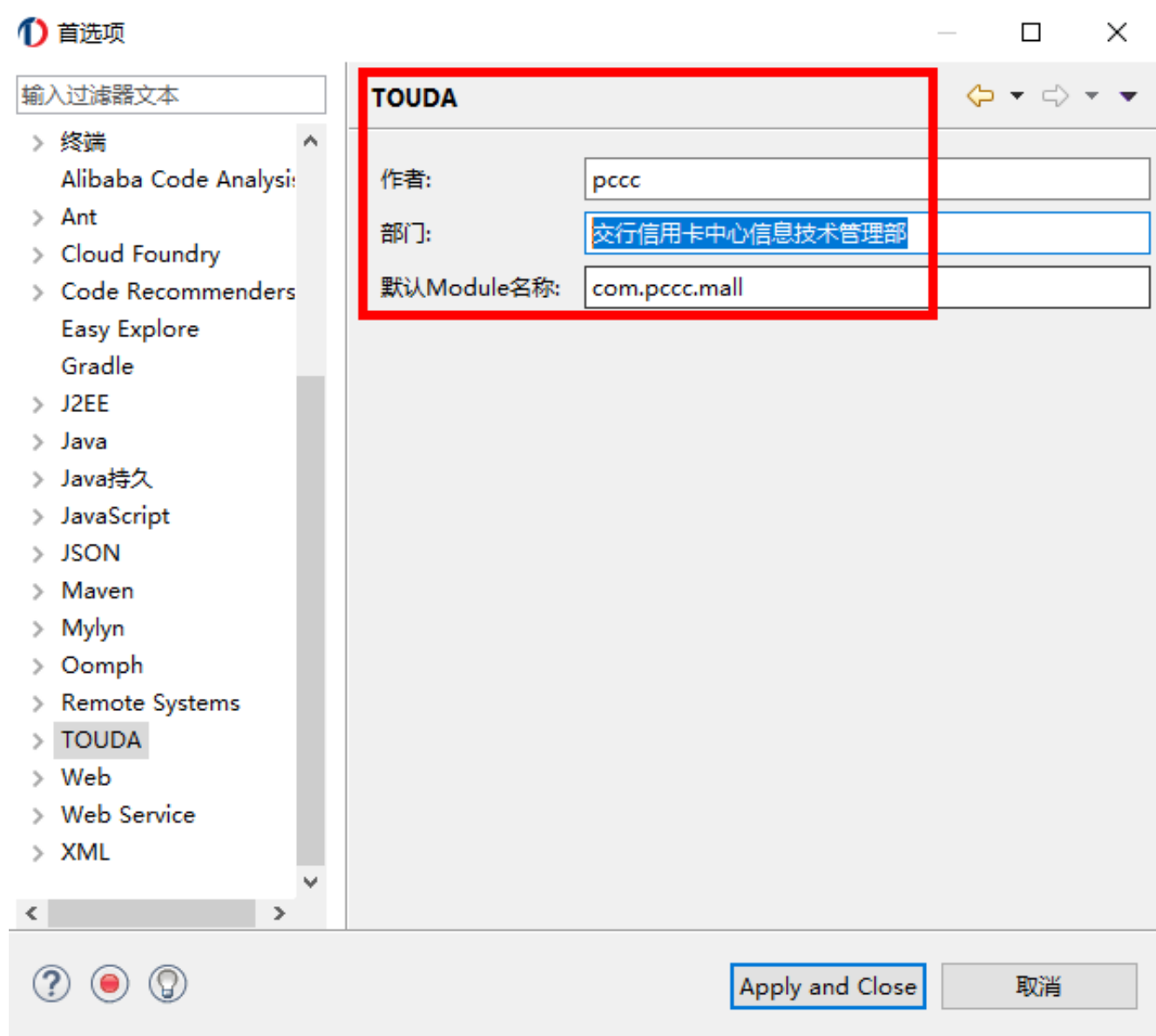
# TOUDA开发规范目录

- IDE配置规范/项目创建规范
- 日志文件及日志规范
- 参数配置规范
- 服务发布流程
- 微应用申请流程

# IDE配置规范/项目创建规范

- Primeton EOS设置（3.0更名为TOUDA设置）
  - 作者：工号/你的姓名
  - 部门：交行信用卡中心信息技术管理部
  - 默认module名称：com.pccc.子项名
- 工作空间编码设置为UTF-8
- JRE1.7以上(包含1.7)
- 编译一致性6.0
- 3.0不需再导入function项目

# IDE配置规范/项目创建规范



# 日志文件规范-1

- 6个日志文件
  - touda-system.log: 边界日志。大粒度的引擎运行的入口、出口的日志，主要记录渠道、通道的请求报文及响应报文信息
  - touda-trace.log: 记录Server引擎的调试日志，用于系统维护人员定位系统运行问题使用，也包括“com.pccc.touda”包下的所有日志，默认是info级别
  - touda-proxy.log: 记录“com.pccc.touda.proxy”包下的日志信息，默认是Info级别，主要用于查看通过Invoke调用服务的日志信息
  - touda-engine.log: 记录细粒度的引擎运行日志，可以打印上下文数据，用于定位业务问题也包括执行的sql脚本情况，数据库连接，执行sql等
  - touda-app.log: 记录开发人员输出的日志
  - touda-deploy.log: 记录部署日志
- 日志级别(不区分大小写):TRACE<DEBUG<INFO<WARN<ERROR
- 只能用TOUDA提供的API,不允许第三方日志框架

# 日志文件规范-1-云上云下不同之处

原云下的日志文件	日志前缀
touda-system.log	syslog
touda-trace.log	tracelog
touda-proxy.log	proxylog
touda-engine.log	engine
touda-app.log	applog
touda-deploy.log	deploylog

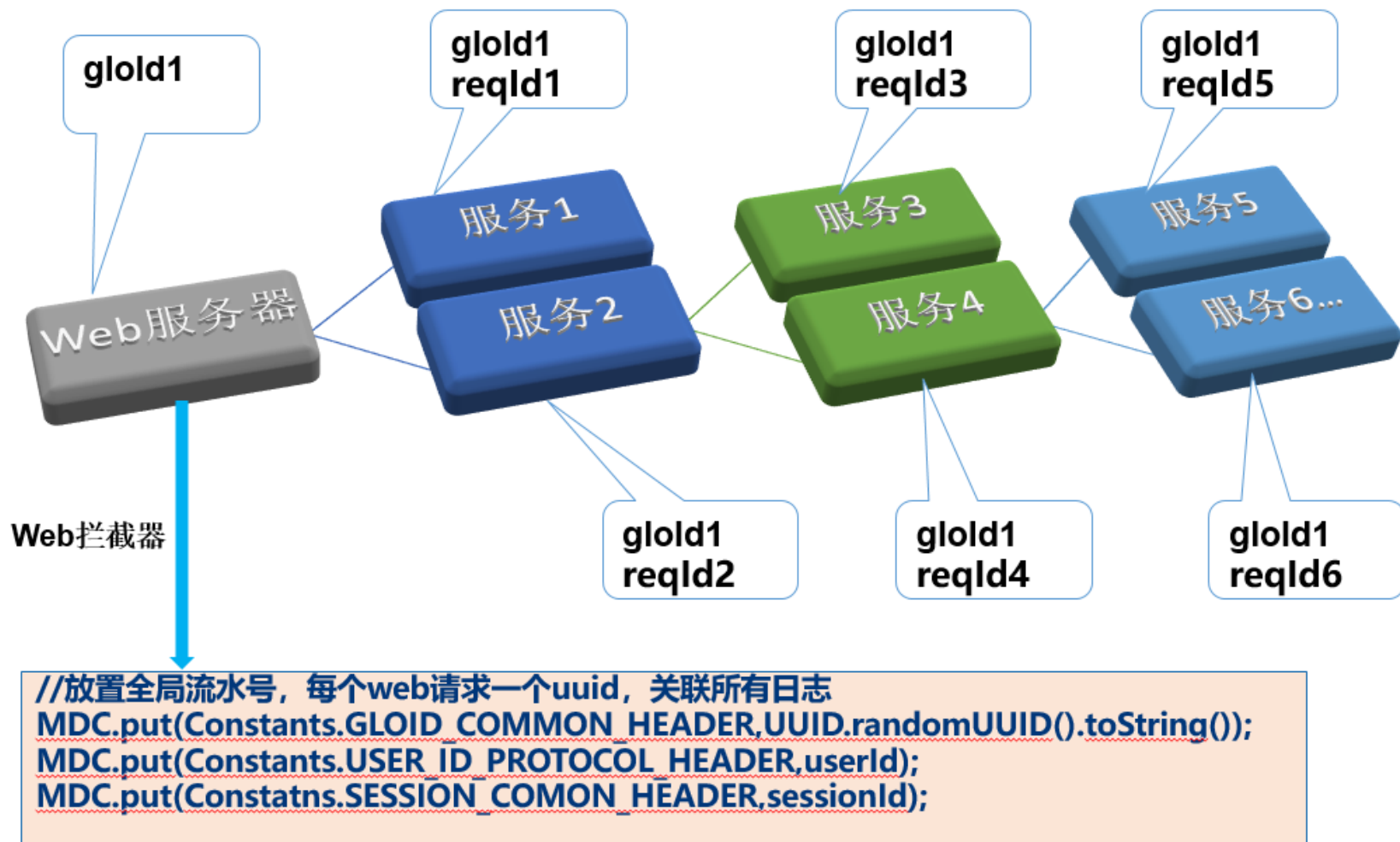
# 日志文件规范-2

- 日志全部输出到KAFKA,由日志平台统一收集处理，并提供查询功能。需要与配置项 **【kafkaAppender.services+kafkaAppender.topic】** 结合使用
- 在WEB项目中需要配置MDC，增加全局流水号以便在日志平台排查问题

```
//放置全局流水号，每个web请求一个uuid，关联所有日志  
MDC.put(Constants.GLOID_COMMON_HEADER,UUID.randomUUID().toString());  
MDC.put(Constants.USER_ID_PROTOCOL_HEADER,userId);  
MDC.put(Constants.SESSION_COMON_HEADER,sessionId);
```



# 日志文件规范-3-全局流水号释义



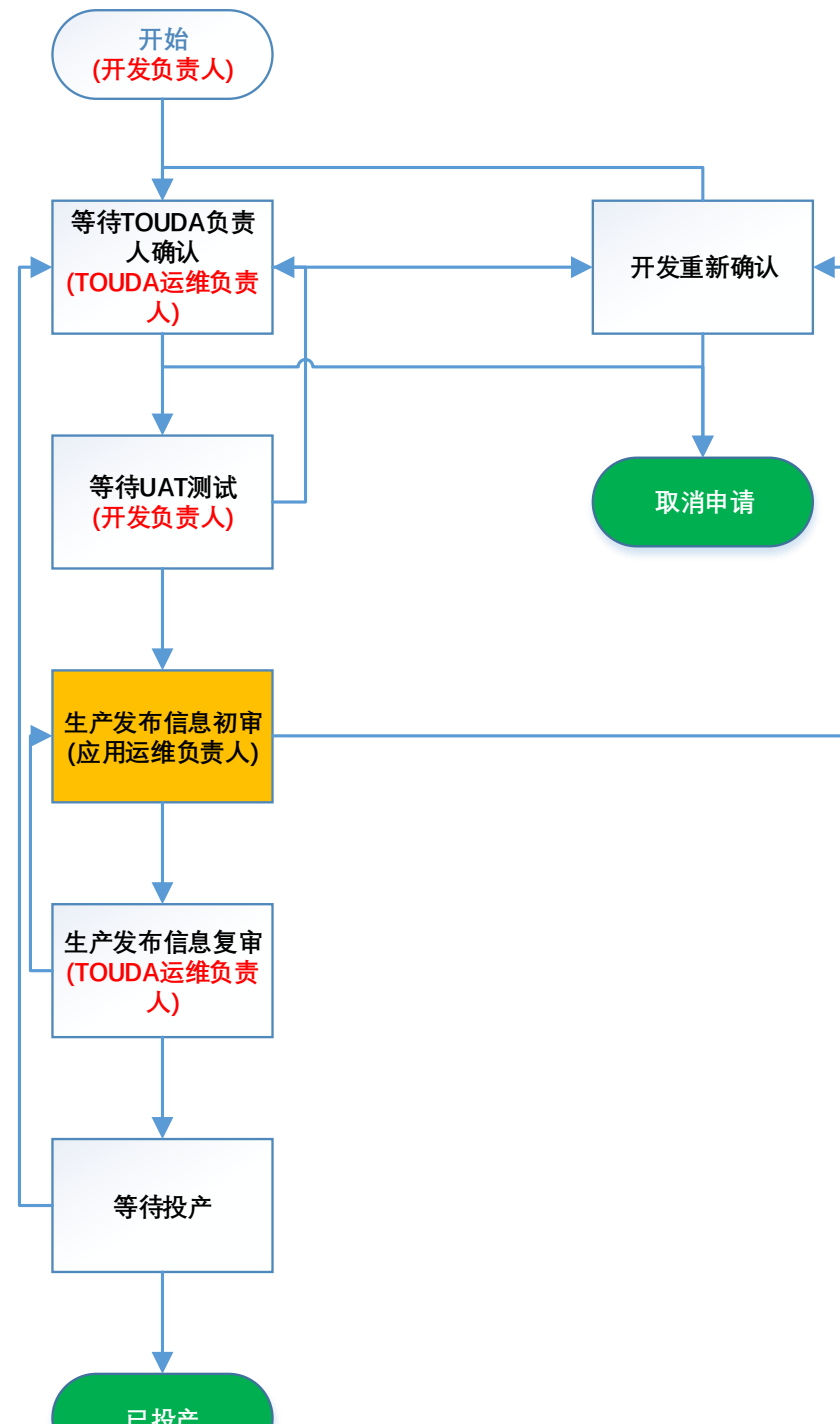
# 参数配置规范

- 见上文配置中心/touda.properties说明

# (必考) 服务发布流程

1. **初始**: 开发完成服务发布申请相关模板, 提交申请
2. **等待Touda负责人确认**: Touda运维负责人审核开发提交的excel模板, 若无误, 则录入到uat环境的服务治理中心, 否则, 驳回
3. **等待uat测试**: 服务信息已经录入到uat环境, 开发进行非直连方式代码验证
4. **生产发布信息初审**: 应用运维负责人负责审核申请表, 补充生产环境信息
5. **生产发布信息复审**: Touda运维负责人审核各个组运维人员补充生产信息后提交的申请单
6. **等待投产**: Touda运维负责人在制定时间根据excel模板将服务录入到生产服务治理中心. 上线前一定需要将流程点到这个状态.
7. **已投产**: 服务已经录入到生产服务治理中心

# (必考) 服务发布流程



# 服务发布申请表-表单填写-profile

- ~~01-Touda应用新增申请表.docx~~
- 02-Touda服务发布申请表.xlsx
- 03-非Touda服务调用申请表.xlsx
- 04-Touda服务变更申请表.xlsx
- 05-Touda服务接入申请表.xlsx

# 服务发布流程-表单填写-01表

- 01表-全名01-Touda应用系统新增申请表.doc
- 该表目前在3.0被【微应用申请流程取代】，不再做填写

# （必考）服务发布流程-表单填写-02表

- 02表-全名02-Touda服务发布申请表.xlsx
- 在开发完TOUDA服务之后需要发布到注册中心供消费调用时填写。

- 1. 应用ID（必填）//服务提供者应用ID，等于touda.app.id
- 2. 消费者应用ID（必填）
- 3. 服务名称（必填）//服务全名,英文名称
- 4. 服务描述（必填）//服务中文名称
- 5. 上UAT时间（必填）
- 6. 上生产时间（必填）
- 7. 版本（正确填写或者留白）
- 8. 缓存KEY
- 9. 失效时间（秒）
- 10. 容错值
- 11. 负责人（卡中心IT部人员）
- 12. 扩展属性
- 13. 请求超时时间
- 14. 备注

		A	B	C	D	E		
		应用ID（必填）	服务名称（	服务描述（必填）		消费者应用ID（必填）		
		pms_mms_service	com.pccc.pms.mms	文宣种类新增		pms_mms_web		
F	G	H	I	J	K	L	M	N
缓存KEY	失效时间（秒）	容错值	负责人	扩展属性	请求超时时	备注	上UAT时间（必填）	生产时间（必填）
			卡中心IT部人员				2017/2/20	2017/4/20

# 服务发布流程-表单填写-03表

- 03表-全名03-非Touda服务调用申请表.xlsx
- 在需要调用非TOUDA开发的服务时填写
- 含有2个sheet
  - 服务调用申请sheet:与02表表头一样
  - 服务提供信息sheet:表头3列（应用ID/服务名称/访问URL），都是必填.特别注意的是若服务提供者提供的访问URL若存在多个，需多行填写，不要写在一行。



# 服务发布流程-表单填写-04表

- 04表-全名04-Touda服务变更申请表.xlsx
- 有三种变更类型
  - 消费者变更
  - 超时时间变更
  - 交易缓存变更
- 剩下的字段与02表一样，即比02表多1个变更类型字段
- 若新增消费者，则在原消费者的后面用逗号分隔追加

# 服务发布流程-表单填写-05表

- 05表-全名05-Touda服务接入申请表.xlsx
- 需要调用别的系统服务时填写
- 表头内容与02表一样，必填项也一样

# (必考) Touda服务申请常见问题

1. 提供者/消费者应用名填写不正确
2. 申请的Touda服务的服务名与真实代码的路径不匹配
3. 服务版本号随意填写(正确填写或者留白都行)
4. 申请的静态服务有以下问题
  - a. 提供者应用名随意填写
  - b. 服务名不是代码中主机服务的服务名(name属性的值)
  - c. 第二个sheet页的URL与注册中心已录入的提供者IP不相符
  - d. 一个服务由多个IP提供,没有分行写(拒绝"合并单元格"以及多个URL写一格)
  - e. 提供者的IP有F5的地址
5. 服务可接入申请时,提供者和服务名与已经上线的服务不相符,要么提供者应用名有误,要么服务名不正确
6. 上UAT时间 (应提前1周) ,上生产时间 (应提前2周) 随意填写(正确时间格式样例:2018/12/05)

# 微应用申请流程-双速流程

- 保障应用名全局唯一












TOUDA-IDE

# TOUDA-IDE 目录

- 开发前
- 开发中
- 开发后

# TOUDA-IDE-开发前-了解IDE目录结构

- ▼  mall\_markert\_service [branches/20181103/services/piap\_mall\_markert\_servi
  - >  架构设计
  - >  业务建模
  - >  服务定制
  - >  EOS构件包
  - >  WEB
  - >  系统配置
  - >  Java库
  -  system.sdvx 184 18-5-14 下午8:48 qy\_dengjiixin

# TOUDA-IDE-开发前-了解IDE目录结构

- 架构设计
  - 包名:com.pccc.子项名.应用名.**architecture**
  - 作用:定义本应用的**渠道/通道/核心的配置文件**(touda.properties/spring-proxy.xml/以及分段式处理的服务器参数配置,详见下文章节)
- 业务建模
  - 包名:com.pccc.子项名.应用名.**businessmodel**
  - 作用:定义本应用的**数据字典/输入输出报文/数据类型**
- 服务定制
  - 包名:com.pccc.子项名.应用名.**servicecustom**
  - 作用:定义本应用的核心业务逻辑,包括**渠道服务/交易服务/事件服务/交易流/本地处理/命名SQL的声明**
- EOS构件包(TOUDA3.0版本改名为服务逻辑)
  - 包名: com.pccc.子项名.应用名.**servicedisplay**
  - 作用:非以上逻辑部分会放在这里
- *TOUDA3.0版本的IDE创建新工程,以上4个构建包将默认创建*



# TOUDA-IDE-开发前-profile-3-开发流程

- 1、创建渠道（一般创建工程时由负责人创建好渠道，8080端口）；
- 2、根据服务的输入输出定义报文（定义数据字典）；
- 3、创建渠道服务；
- 4、创建交易服务；
- 5、创建交易流；
- 6、完成业务逻辑。

# TOUDA-IDE-开发前-1

- 见[IDE配置规范/项目创建规范](#)

# TOUDA-IDE-开发前-2

- 命名规范总结

- 包名: com.pccc.子项名.xxxx

- 交易码:

- 业务域(三个字符简称, 简称不得超过5位)+\_+模块简称(三位数字)+\_+功能简称(四个数字), 使用有意义的英文单词的简写, 英文单词头字母为大写, 其他为小写, 字符长度不能超过20个字符, 如: 公共模块订单查询交易=PUB\_001\_0001

- 交易码不能重复

- 以上例子, 以交易码为核心对应起来, 则对应的项目分别如下:

- 渠道: HTTP\_PUB\_01

- 渠道服务分类(文件夹): HTTP\_PUB

- 渠道服务: HTTP\_PUB\_001\_0001

- 交易服务: TS\_PUB\_001\_0001

- 交易流: PUB\_001\_0001

- 扩展实现类的结尾: 以对应的中英文名称判断即可

- 其他命名规范参考前文关于TOUDA概念释义章节

# TOUDA-IDE-开发前-2-命名规则参考表

原部件名	构建包	结尾名	扩展实现类所在包名
渠道	架构设计	Transport	com.pccc.子项名.应用ID.architecture.transport
通道	架构设计	Endpoint	com.pccc.子项名.应用ID.architecture.endpoint
数据转换(只有渠道通道才有)	架构设计	DataExchange	com.pccc.子项名.应用ID.architecture.endpoint.dataexchange
渠道服务	服务定制	TransportService	com.pccc.子项名.应用ID.servicecustom.transportservice
交易服务	服务定制	TransService	com.pccc.子项名.应用ID.servicecustom.transservice
主机服务	服务定制	HostTran	com.pccc.子项名.应用ID.servicecustom.hosttrans

拦截器类型	结尾名
Receive类型拦截器	Receive
Send类型拦截器	Send
Request类型拦截器	Request
注册拦截器类	Launch

# TOUDA-IDE-开发前-3-多数据源配置

- 修改user-config.xml文件
- 修改contribution.eosinf
- 修改配置中心（测试环境），
- 数据库密码自动加密

```
<group name="mall_base">
  <configValue key="C3p0-DriverClass">com.ibm.db2.jcc.DB2Driver</configValue>
  <configValue key="C3p0-Ur1">jdbc:db2://182.180.198.242:55000/P01_DB</configValue>
  <configValue key="C3p0-UserName">p01_user</configValue>
  <configValue key="C3p0-Password">{3DES}XJoem4zcUy710gDuGnbpFyaaqYaTx+xtZA==</configValue>
  <configValue key="C3p0-PoolSize">10</configValue>
  <configValue key="C3p0-MaxPoolSize">50</configValue>
  <configValue key="C3p0-MinPoolSize">10</configValue>
  <configValue key="Transaction-Isolation">ISOLATION_READ_COMMITTED</configValue>
  <configValue key="Database-Type">DB2</configValue>
  <configValue key="Jdbc-Type">IBM DB2 Driver(Type4)</configValue>
  <configValue key="Test-Connect-Sql">SELECT count(*) from MALL_PRODUCTS</configValue>
  <configValue key="Retry-Connect-Count">-1</configValue>
</group>
```

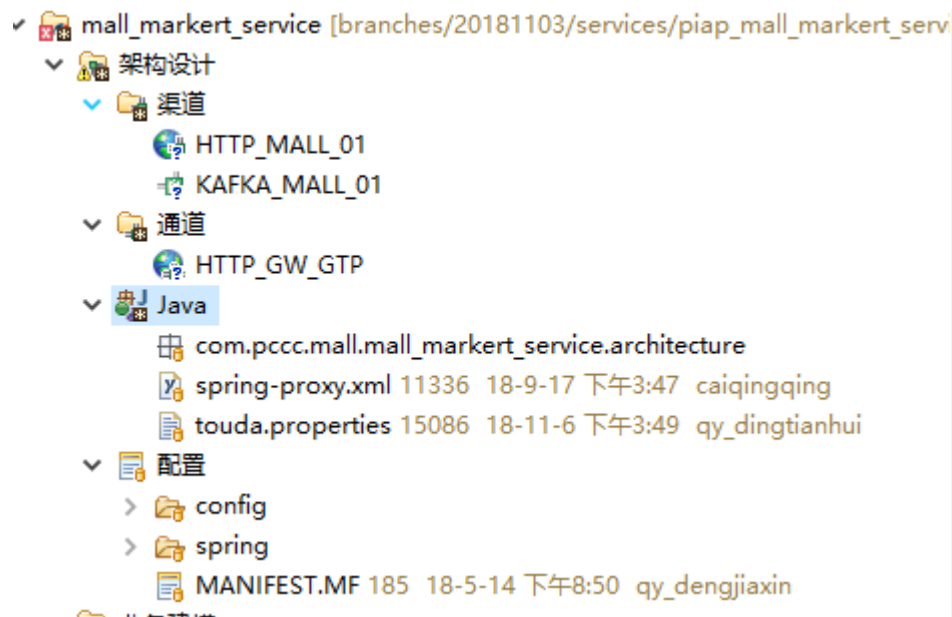
contribution.eosinf

```
23 <!--
24     the configuration below describes
25     the corresponding relationship between contribution datasources
26     multiple datasources can be defined.
27     the value 'default' of attribute 'key' denotes a contribution datasource
28     and the field value 'default' of 'configValue' node stands for an a
29     -->
30     <configValue key="mall_base">mall_base</configValue>
31     <configValue key="mall_tx">mall_tx</configValue>
32   </group>
33 </module>
34 </contribution>
35
```

```
DATABASE_GROUP_NAME_1 = mall_base##1
DATABASE_URL_1 = jdbc:db2://182.180.198.242:55000/P01_DB##1
DATABASE_USERNAME_1 = p01_user##1
DATABASE_PASSWORD_1 = U2FsdGVkX1//owDG2LZ7j7+Xmrqc9dE7283176M==##1
DATABASE_GROUP_NAME_2 = mall_tx##1
DATABASE_URL_2 = jdbc:db2://182.180.198.243:55000/P01_DB##1
DATABASE_USERNAME_2 = p01_user##1
DATABASE_PASSWORD_2 = U2FsdGVkX187ZVW7yoUq2RbCSFjyD0uxkyDOGvM==##1
```

# TOUDA-IDE-开发时-1-架构设计构建包

- 注意渠道/通道命名规范
- HTTP端口号8080，避开9090
- HTTP渠道中主机名配置项必须是0.0.0.0
- 目录结构如下

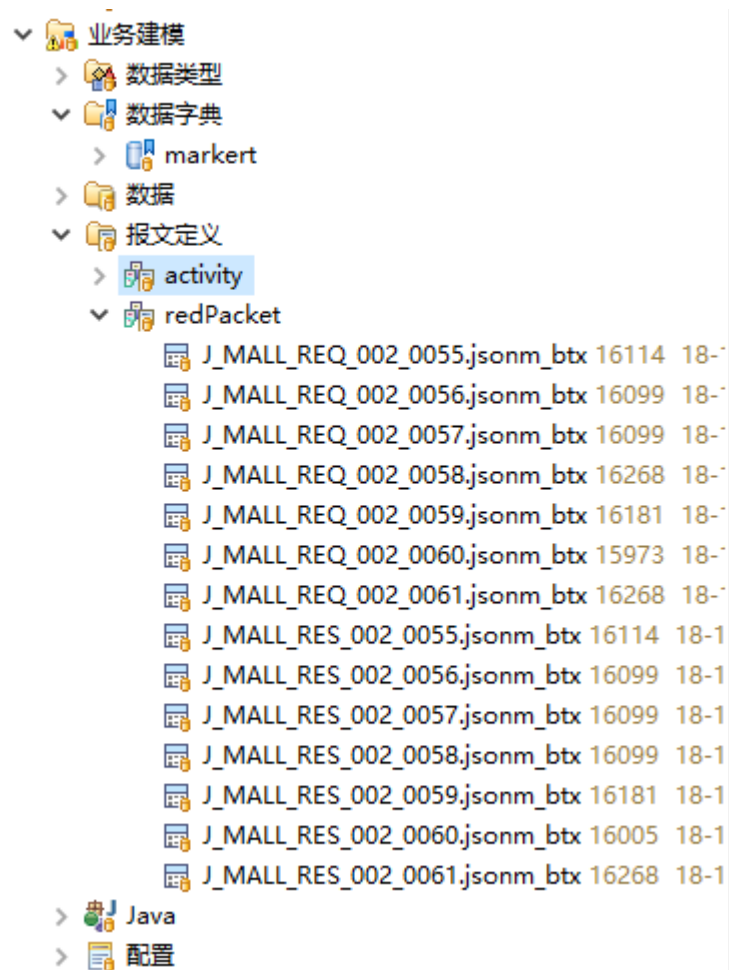


# TOUDA-IDE-开发时-1-HTTP渠道

- 参考上文章节[TOUDA-渠道](#)

# TOUDA-IDE-开发时-2-业务建模构建包

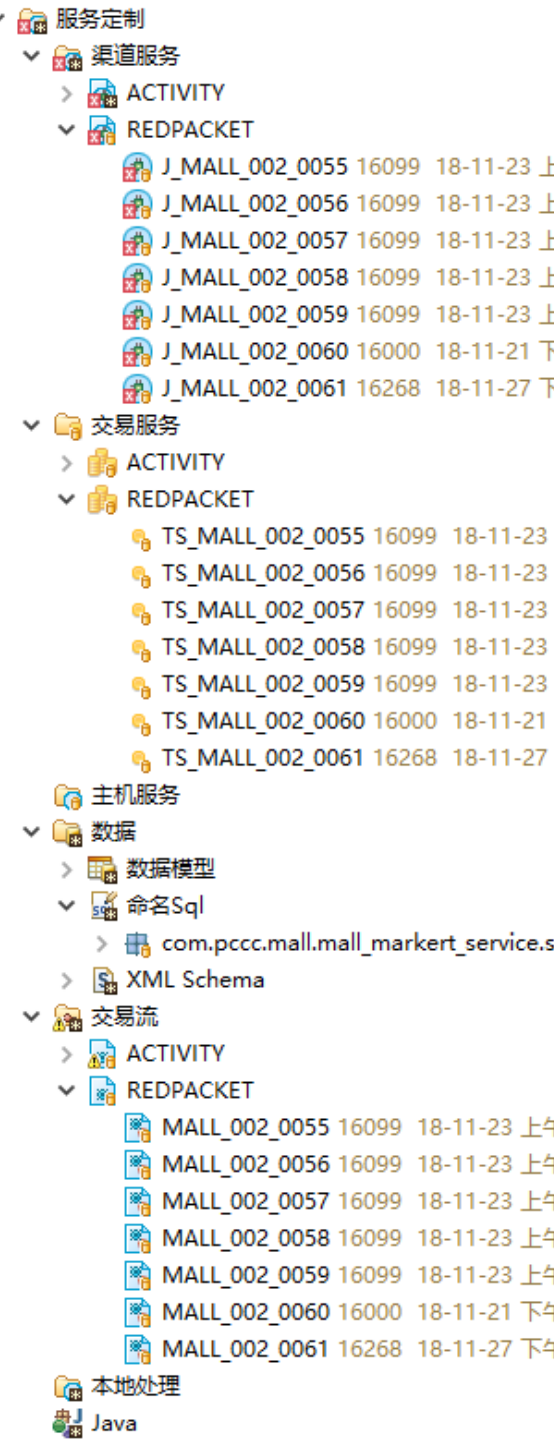
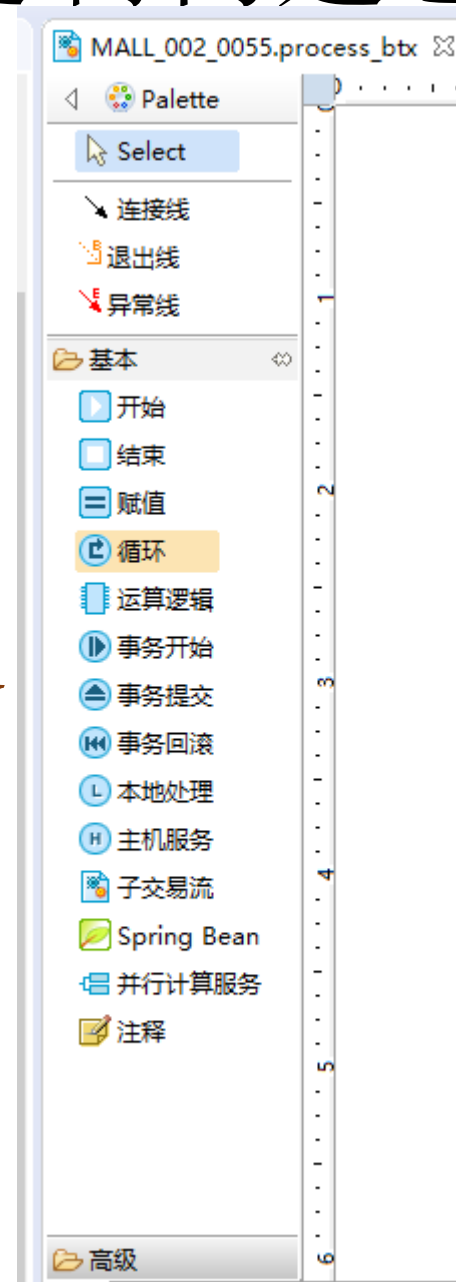
- 数据字典命名META\_DICT
- 报文命名规范
- Touda内置数据类型





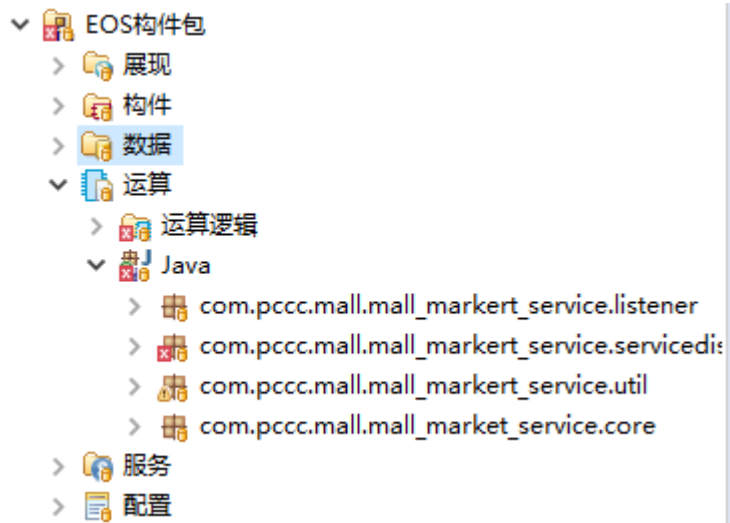
# TOUDA-IDE-开发时-3-服务定制构建包

- 渠道服务/交易服务/交易流的关系
- SQL一般写在这
- 认识交易流图元
  - 连接线、退出线、异常线
  - 开始、结束、注释
  - 赋值、循环、并行计算
  - 运算逻辑、本地处理、子交易流、主机服务
  - 事务开始、提交、回滚
  - SpringBean
- 主机服务不是必须的
- 命名规范



# TOUDA-IDE-开发时-4-EOS构件包

- 3.0版本名字叫服务逻辑
- 一些其他的实现会放在这里



# TOUDA-IDE-开发时-常见配置文件总结

- TOUDA服务端

- spring-proxy.xml(服务代理配置)
- touda.properties(测试环境配置中心)
- user-config.xml(数据源配置,测试环境需要依赖配置中心)
- contribution.eosinf(与user-config.xml必须配合使用)
- logback.xml
- DefaultEventServer.evsr\_btp(事件服务器配置)
- DefaultHostTransServer.htsr\_btp(主机服务器配置)
- DefaultInboundRoute.inrt\_btp(渠道服务路由配置)
- DefaultInboundServer.insr\_btp(渠道服务服务器配置)
- DefaultOutboundRoute.outrt\_btp(主机服务路由配置)
- DefaultReversedTransServer.rssr\_btp(冲正交易服务器配置)
- DefaultTransServer.txsr\_btp(交易服务器配置)

- TOUDA客户端(指任何需要调用TOUDA服务的应用)

- spring-proxy.xml(服务代理配置)
- touda.properties(测试环境配置中心)

# TOUDA-IDE-开发时-5-EventBus

- 不仅仅可用在ToudaIDE里面，只要依赖相关google相关jar都可使用
- 基于发布/订阅模式的事件总线
- 代码可读性好,耦合性低
- 步骤
  1. 首先需要定义一个消息类，该类可以不继承任何基类也不需要实现任何接口
  2. 注册事件。
  3. 创建消息对象，产生事件，即发送消息。
  4. 处理消息。

# TOUDA-IDE-开发时-EventBus示意

//注册事件

```
EventManager.INSTANCE.registerAsynProcessor(new PushOrderDelayJobHandler());
```

//发布消息

```
OrderMessage message = new OrderMessage();  
EventManager.INSTANCE.postAsynEvent(message);
```

//消费消息

@Subscribe

@AllowConcurrentEvents

```
public void handle(OrderMessage orderMessage) {  
    Map<String, Object> params = new HashMap<>();
```

# TOUDA-IDE-开发时-6-注意事项

1. 只允许创建一个HTTP渠道，且避开9090端口,上云8080
2. 渠道服务业务码3-5位
3. 渠道服务的描述(=接口注释)必须填写，描述信息-查询/修改/新增/删除，如：用户信息查询-查询
4. 查询类、单条sql语句及无事务要求的交易，必须选择无事务、全部交易禁止选择全局事务
5. 若交易流中自己控制事务，则交易服务中需选择无事务
6. 所有服务的响应报文中必须包含returnCode(000000=成功)和returnMsg

# TOUDA-IDE-开发后

- TOUDA服务发布流程，请参考[（必考）服务发布流程](#)
- 部署流程（PaaS云章节补充）
- 他系统如何调用服务，请参考[TOUDA-服务代理](#)